Belirtilere Dayalı Hastalık Tahminine ve Bölüm Yönlendirmesine Yapay Zekâ Yaklaşımı

Artificial Intelligence Approach to Symptom-Based Disease Prediction and Department Routing

Tuğberk DİLAVER ve Tülay YILDIRIM
Elektronik ve Haberleşme Mühendisliği
Yıldız Teknik Üniversitesi
İstanbul, Türkiye
tugberk.dilaver@gmail.com, tulay@yildiz.edu.tr

***Özetçe*—Günümüzde hastanelerde kullanılan randevu alma sistemlerinde hastalar, hangi hastalığa sahip olduklarına dair bir fikir sahibi olmadan randevu almaktadır. Yanlış randevu alınması hem hastayı hem de hastaneyi zaman ve para konusunda olumsuz etkileyebilme potansiyeline sahiptir. Buna ek olarak hastalar, sahip oldukları belirtilerin hangi hastalığın belirtileri olduğunu ve hastalık için hastanelerin hangi bölümünün hizmet verdiğini genellikle net bir şekilde bilememektedir. Bu çalışmada, kullanıcılara ön tanımlı olarak 132 farklı hastalık belirtisi arasından sahip olduklarını seçme hakkı tanıyarak, kullanıcının 41 farklı hastalık arasından hangi hastalıklara sahip olduğu, Multinomial Naïve Bayes algoritması ile oluşturulmuş yapay zekâ modeli aracılığıyla tahmin edilmektedir. Çalışmada, React kütüphanesi ile oluşturulmuş bir arayüzde kullanıcıya, seçmiş olduğu belirtilere göre en yüksek olasılığa sahip ilk 3 hastalık tahmini gösterilmektedir. Bunun yanı sıra, hastalıkların hastanelerdeki ilgili bölümü ve açıklamalarının yer aldığı bilgiler sunularak, kullanıcının bilgilendirilmesi hedeflenmiştir.**

Anahtar Kelimeler — Yapay zekâ; veri seti; makine öğrenmesi; hastalık tahmini; Multinomial Naïve Bayes algoritması; React

***Abstract*—In contemporary hospital appointment systems, patients often book appointments without having any idea about the nature of their ailment. Incorrect appointment bookings have the potential to negatively impact both the patient and the hospital in terms of time and cost. Additionally, patients are often unclear about which symptoms correspond to which diseases and which hospital departments provide relevant services for their conditions. In this study, by giving users the right to choose among 132 predefined different disease symptoms, the diseases the user has among 41 different diseases are predicted through an artificial intelligence model created with the Multinomial Naïve Bayes algorithm. In the study, the user is presented with the top three disease predictions with the highest probability based on the symptoms selected in an interface developed with the React library. Additionally, the study aims to inform the user by providing information about the relevant hospital departments and explanations of the diseases.**

Keywords — Artificial intelligence; dataset; machine learning; disease prediction; Multinomial Naïve Bayes algorithm; React

979-8-3315-2981-9/24/$31.00 ©2024 IEEE

I. GİRİŞ

Yapay zekâ teknolojilerinin günümüzde yaygınlaşması ve uygulama alanlarının genişlemesiyle birlikte bu teknoloji birçok farklı sektörde kullanılmaya başlamıştır ve bu konuda yapılan araştırma sayısı da her geçen gün artmaktadır. Yapay zekâ konusunda araştırma yapılan alanların en önemlilerinden biri de sağlık sektörü olarak karşımıza çıkmaktadır ve bu alanda yapılmış olan birçok çalışma mevcuttur.

Yapay zekânın sağlık alanında ilk kullanıldığı proje 1972 yılında Stanford Üniversitesi'nde enfeksiyonlara neden olan bakterilerin saptanması ve antibiyotik önerilmesi üzerine olan MYCIN projesidir [1].

Günümüze yakın tarihlerdeki yapay zekâ ile hastalık tahmini üzerine yapılan akademik çalışmalar incelendiğinde hastalık belirtileri üzerinden hastalık tahmini yapan çalışmalar arasında Dahiwade, Patle ve Meshram [2], K-En Yakın Komşu (KNN) ve Evrişimli Sinir Ağları (CNN) algoritmalarını kullanmışlardır. Çalışmalarında CNN algoritması ile KNN'e göre daha yüksek bir oranda doğruluk elde etmişlerdir.

Bir diğer çalışmada, Kumar, Sathya, Nadeem ve Rajesh [3], geliştirdikleri grafiksel kullanıcı arayüzü üzerinden seçilen en az 2 adet belirtiye göre hastalık tahmini yapmıştır ve Rastgele Orman algoritması ile oluşturdukları modelde %95.12 doğruluk oranı elde etmişlerdir. Kullandıkları veri setinin bizim çalışmamızda olduğu gibi 41 hastalık içerdiği görülmüştür.

Bir diğer çalışmada ise Grampurohit ve Sagarnal [4], 41 hastalık türü için belirtilere dayanarak Karar Ağaçları, Rastgele

Orman ve Naïve Bayes (NB) algoritmaları ile hastalık tahmini üzerine oluşturdukları yapay zekâ modellerinin karşılaştırmalı sonuçlarına yer vermişlerdir ve kullandıkları veri setinin bizim çalışmamızda kullanılan ile benzer olduğu görülmüştür.

Sağlık alanında yapılmış ve son kullanıcıya ulaşmış Türkiye Cumhuriyeti Sağlık Bakanlığı'nın bir uygulaması olan "NeyimVar" uygulaması bu konuda dikkat çeken bir diğer çalışmadır [5]. Uygulama, hastanın sahip olduğu belirtilere göre uygun branşa hastanın yönlendirilmesini sağlamaktadır.

II. Veri Seti Analizi

Bu çalışmada veri seti için Kaggle web sitesinden yararlanılmıştır [6]. 2010 yılında kurulmuş olan Kaggle web sitesi Google LLC bünyesinde faaliyet gösteren veri bilimi ve makine öğrenimi ile ilgili çevrimiçi bir topluluktur. Web sitesi üzerinde çeşitli kategorilerde veri setleri yer almaktadır.

Uygulamada, Kaggle web sitesi üzerinde "Disease Symptom Prediction" isimli bir veri seti paylaşımından yararlanılmıştır [7]. Bu veri setinde "dataset", "symptom_description", "symptom_precaution" ve "symptom_severity" olmak üzere 4 adet dosya bulunmaktadır ve dosyalar Virgülle Ayrılmış Değerler yani "Comma Separated Values" (CSV) formatındadır. Çalışma kapsamında "dataset" isimli dosya ana veri kaynağıdır. İlgili dosya içerisinde 41 hastalık türü için 4920 kayıt ve her hastalık türü için 120 kayıt bulunmaktadır. Veri setinde yer alan 41 hastalık Tablo I'de gösterilmiştir.

TABLO I. Veri Setinde Yer Alan Hastalıklar

Hastalıklar			
AIDS	Hepatit B	Migren	Soğuk Algınlığı
Akne	Hepatit C	Osteoartrit	İlaç Reaksiyonu
Alerji	Hepatit D	Sarılık	Servikal Spondiloz
Alkolik Hepatit	Hepatit E	Sedef	İdrar Yolu Enfeksiyonu
Artrit	Hipertansiyon	Sıtma	Kronik Kolestaz
Dang Humması	Hipertiroidizm	Suçiçeği	Mantar Enfeksiyonu
Diyabet	Hipotiroidizm	Tifo	Gastroözofageal Reflü
Gastroenterit	Hipoglisemi	Varis	Paroksismal Pozisyonel Vertigo
Peptik Ülser	Hemoroid	Verem	Bronşiyal Astım
Kalp Krizi	İmpetigo	Zatürre	Felç (Beyin Kanaması)
Hepatit A			

Veri seti kayıtlarında bir adet "Disease" ve 1'den 17'ye kadar numaralandırılmış 17 "Symptom" belirti sütunu ile birlikte toplamda 18 sütun bulunmaktadır. Bu 17 belirti sütununda 132 farklı belirtiye yer verilmiştir. Veri setinde yer alan belirtiler "cold_hands_and_feets" şeklinde, kelimeler arasında "_" olacak şekilde yer almaktadır. Tablo II'de veri setinde yer alan ilk 3 kayıt gösterilmiştir.

TABLO II. Veri Setinde Yer Alan İlk 3 Kayıt

Disease	**Symptom_1**	**Symptom_2**	**Symptom_3**	**...**
Fungal infection	itching	skin_rash	nodal_skin_eruptions	...
Fungal infection	skin_rash	nodal_skin_eruptions	dischromic_patches	...
Fungal infection	itching	nodal_skin_eruptions	dischromic_patches	...

III. Algoritma Seçimi

Veri seti temelde etiketlenmiş veriler içermektedir. Veri setinde etiketler hastalıklar olurken, etiketlenen veriler ise belirtilerdir. Ancak veri seti incelendiğinde 17 adet belirti/semptom sütunu yer almasına rağmen, her belirti sütununun benzer ifadeleri ve değerleri içerebildiği de gözlemlenmiştir.

Veri setinin içerisinde sürekli değişen bir veri olmadığından ve bir sınıflandırma işlemine ihtiyaç duyulduğundan, en uygun algoritma türü olarak sınıflandırma işlemini gerçekleştirmek için sıklıkla kullanılan Naïve Bayes (NB) algoritması tercih edilmiştir.

NB, doğal dil işlemede ve metin tabanlı veri setlerinde yaygın olarak kullanılan olasılığa dayalı bir sınıflandırma makine öğrenimi algoritmasıdır. Bu algoritmanın amacı, metindeki kelimeler verildiğinde, bir metin parçasının belirli bir sınıfa ait olma olasılığını hesaplamaktır.

NB algoritması, semptomların yani belirtilerin hastalıklara ait olma olasılığını (1) ile hesaplar.

$$P(H|B)=\frac{P(B|H)*P(H)}{P(B)} \quad (1)$$

P: Olasılık

H: Hastalık

B: Belirti

P(H|B): Belirtiler var olduğunda, hastalığın olma olasılığıdır. Bu, tahmin edilmek istenen değerdir.

P(B|H): Hastalık var olduğunda, belirtilerin ortaya çıkma olasılığıdır. Bu, belirli bir hastalığa sahip olan bireylerin bu belirtileri gösterme olasılığıdır.

P(H): Veri setinde hastalığın görülme olasılığıdır.

P(B): Veri setinde belirtilerin herhangi bir hastalığa bağlı olmaksızın genel görülme olasılığıdır.

Bu formül birden fazla belirtiyi içerecek şekilde genelleştirildiğinde (2) elde edilir:

$$P(H|B_1,B_2,\ldots,B_n)=\frac{P(B_1|H)*P(B_2|H)*\ldots*P(B_n|H)*P(H)}{P(B_1,B_2,\ldots,B_n)} \quad (2)$$

NB algoritmasının Multinomial, Bernoulli ve Gauss gibi farklı türleri bulunmaktadır. Bu çalışmada Multinomial Naïve Bayes (MNB) algoritması tercih edilmiştir. Diğer türlerinin aksine metindeki her kelimenin belirli bir sınıfta ne kadar sık geçtiğini dikkate almasından ve kelime frekanslarını kullanarak

sınıflandırma yapmasından dolayı bu algoritma tercih edilmiştir.

MNB algoritmasının sınıflandırma performansını arttırmak için metin verilerini sayısal değerlere dönüştürmeyi sağlayan ve metin içindeki ifadelerin önemini gösteren "Term Frequency-Inverse Document Frequency" (TF-IDF) yöntemi kullanılmıştır.

TF-IDF, bir metindeki kelimelerin önemini belirlemek için "Term Frequency" (TF) ve "Inverse Document Frequency" (IDF) olmak üzere iki ölçümü birleştirir. TF, bir dokümandaki bir ifadenin sıklığını ölçmektedir. Doküman içerisinde sık geçmekte olan ifade ya da kelimeler daha yüksek TF değerine sahip olmaktadır. TF, (3) ile hesaplanmaktadır:

$$TF(k,d)=\frac{\text{d'deki k'nin sayısı}}{\text{d'deki toplam kelime sayısı}} \quad (3)$$

k: Kelime

d: Doküman

IDF ise, bir ifade ya da kelimenin tüm metinler ya da dokümanlar içerisindeki sıklığını ölçmektedir. Az sayıda metin içerisinde geçen kelimeler daha yüksek IDF değerine sahip olmaktadır. IDF, (4) ile hesaplanmaktadır:

$$IDF(k)=\log\left(\frac{\text{toplam doküman sayısı}}{\text{içinde k geçen doküman sayısı}}\right) \quad (4)$$

k: Kelime

TF-IDF değeri ise (5) ile hesaplanmaktadır:

$$TF\text{-}IDF(k)=TF(k,d)*IDF(k) \quad (5)$$

Hesaplama sonucunda elde edilen değer, belirli bir metin içinde sıkça geçen ancak diğer metinlerde nadir olarak tanımlanabilecek kelimeler olduğunu gösterir.

IV. Veri Seti Ön İşlemleri ve Yapay Zekâ Modelinin Oluşturulması

MNB algoritmasına belirtileri verebilmek için, hastalıkların sahip olduğu belirtiler Tablo II.'de verildiği gibi iken, aralarına birer boşluk yerleştirilerek Tablo III'de olduğu gibi "Symptoms" sütunu altında birleştirilmiştir. Birleştirme işlemi ardından "Symptoms" sütununda yer alan kayıtların başındaki ve sonundaki boşluklar kaldırılmıştır.

TABLO III. Ana Veri Kaynağı Örnek Kayıtları

Disease	Symptoms
Fungal infection	itching skin_rash nodal_skin_eruptions
Fungal infection	skin_rash nodal_skin_eruptions dischromic_patches
Fungal infection	itching nodal_skin_eruptions dischromic_patches

Belirtilerin Tablo III'de gösterildiği şekilde birleştirilmesinin ve diğer işlemlerin ardından Python ile MNB algoritması ve TF-IDF yöntemi kullanılarak model oluşturulmuştur. Model oluşturulurken veri setindeki kayıtların %80'i eğitim ve %20'si ise test için ayrılmıştır.

Tüm işlemlerin ardından oluşturulan modelin doğruluk oranı %100 olarak elde edilmiştir.

V. Uygulama Programlama Arayüzünün Oluşturulması

Bölüm IV'de anlatımı gerçekleştirilen yapay zekâ modeli Flask web çerçevesi ile oluşturulan uygulama programlama arayüzü ya da "Application Programming Interface" (API) içinde kullanılmıştır [8].

API, bir yazılımın başka bir yazılım ile iletişim kurmasına ve birbirleri ile veri alışverişi yapmasına imkân sağlayan bir dizi tanımlanmış kurallar ve protokollerdir. API'ler sahip oldukları bu özelliklerinden ve yazılımların modüler, yeniden kullanılabilir ve de ölçeklenebilir olmasını sağlamalarından dolayı günümüz yazılım projelerinde sıklıkla kullanılmaktadır.

Flask, Python ile web uygulamaları geliştirmek için kullanılan hafif ve kullanımı kolay bir web çerçevesidir. Sahip olduğu kullanım kolaylığı, esneklik, hafiflik ve dokümantasyonlardan kaynaklı olarak projede tercih edilmiştir.

API içerisinde arayüzden gelen istekleri karşılaması için "/predict", "/symptomlist" ve "/diseaseinfo" olmak üzere 3 adet uç noktası kullanılmaktadır.

A. */predict Uç Noktası*

"/predict", kullanıcının arayüzde seçmiş olduğu belirtileri yapay zekâ modelinden geçirerek hastalık tahmini yapan fonksiyonların sonucunu gönderir.

B. */symptomlist Uç Noktası*

"/symptomlist", kullanıcı arayüzünde yer alan açılır çoklu seçim menüsünde kullanılmak üzere oluşturulmuştur. İstek geldiğinde, bir CSV formatlı dosyada yer alan hastalık belirtilerini okur ve bu belirtileri gönderir.

C. */diseaseinfo Uç Noktası*

İlgili API uç noktası, "/predict" uç noktasında tahmini yapılan hastalıklar ile ilgili hastane bölümleri ve hastalık açıklamaları gibi diğer bilgileri CSV formatlı bir dosyadan okuyarak arayüzde göstermek amacıyla oluşturulmuştur.

VI. Uygulama Kullanici Arayüzünün Oluşturulması

Uygulama arayüzünde React kullanılmıştır. React, kullanıcı arayüzleri geliştirmek için kullanılan açık kaynaklı bir JavaScript kütüphanesidir ve Facebook tarafından geliştirilmiştir [9].

React, bileşen tabanlı mimarisi ile kullanıcı arayüzlerini yeniden kullanılabilir küçük ve bağımsız parçalara bölmektedir ve bu özelliğiyle uygulama geliştirme süreçlerini daha verimli hale getirmeye yardımcı olmaktadır.

React'ın sağlamış olduğu en büyük avantajlardan biri, "Virtual Document Object Model" (Sanal DOM) kullanarak kullanıcı arayüzü güncellemelerini hızlı ve verimli bir şekilde gerçekleştirebilme kabiliyetidir.

DOM, bir web sayfasının hiyerarşik bir bileşen ağacıdır ve web sayfasının dinamik olarak düzenlenmesine imkân tanır [10].

Sanal DOM, gerçek DOM'un hafif bir kopyasıdır ve bu sayede React, bileşenlerin durumlarında değişiklik meydana geldiğinde bu iki DOM'u karşılaştırarak bütün kullanıcı arayüzünü yeniden yüklemek ya da oluşturmak yerine sadece değişen parçaları güncellemektedir.

A. *Uygulama Çıktıları*

Kullanıcı web uygulamasına eriştiğinde Şekil 1'de yer alan ekran ile karşılaşmaktadır ve bu ekranda 132 adet belirtinin listelendiği bir açılır menü ve de bir analiz tuşu yer al. Kullanıcı, bu açılır menü üzerinde belirtileri Şekil 1'de gösterildiği gibi çoklu olarak ve herhangi bir sınır olmadan seçebilmektedir.



Şekil. 1. Uygulama ekranında yer alan açılır menü ve içeriği

Kullanıcı, uygulamada belirtileri seçtikten sonra "Analiz Et" tuşuna tıkladığında, seçimi yapılan belirtilerle en yüksek olasılıkla eşleşen ilk 3 hastalık "Analiz Sonucu" başlığı altında en yüksek olasılığa sahip hastalıktan başlanarak Şekil 2'de olduğu gibi listelenmektedir ve "Ek Bilgiler" bölümünde ise tahmini yapılan hastalıklar ile ilgili açıklamalar ve hastane bölümleri gösterilmektedir.



Şekil. 2. Uygulama analiz sonucu

VII. SONUÇLAR VE TARTIŞMA

Yapılan uygulama testlerinde modelin ve uygulamanın başarılı bir şekilde girilen belirtilere göre hastalık sınıflandırması yaptığı görülmüştür. Uygulama kullanıcının girmiş olduğu belirtilere en uygun eşleşmeye sahip hastalıkları ve bu hastalıklar ile ilgili diğer bilgileri de başarılı bir şekilde kullanıcıya göstermektedir.

Elde edilen sonuçlar dahilinde model, TF-IDF vektörleştirici ile metin verilerini sayısal verilere dönüştürerek eğitim ve test işlemlerini başarılı olarak gerçekleştirmiştir. Bu durum, TF-IDF vektörleştirici ve MNB kombinasyonunun, metin sınıflandırma problemlerinde oldukça etkili olabileceğini göstermektedir.

Modelin bu kadar yüksek performans göstermesinin bir nedeni olarak veri setinin iyi etiketlenmiş ve her bir hastalık için aynı sayıda kayıt içermesinden kaynaklandığı düşünülmektedir.

Ancak oluşturulan modelin doğruluğu her ne kadar iyi bir seviyede olsa da modelin test verilerini ezberlemiş olması yani aşırı öğrenme ihtimali olasılıklar dahilinde olduğundan, bu durumu irdelemek adına veri seti daha yakından incelenmiştir.

Yapılan incelemenin ardından içerisinde 4920 kayıt yer alan veri setinde çok sayıda birbirinin aynı olan kayıt yer aldığı görülmüştür. Veri setinde aynı olan kayıtlar çıkarıldığında geriye kalan 304 kayıt ile Bölüm IV'deki yöntemlerle yeni bir model oluşturulmuştur. Yeni oluşturulan modelin doğruluk oranı %85 olarak elde edilmiştir. Ancak yeni oluşturulan modelin 8 hastalık türünü sınıflandıramamasından dolayı uygulamada bir önceki model kullanılmıştır.

Bu sonuçlar dahilinde, ileride daha kapsamlı ve farklı veri setleri ve yapay zekâ modelleri ile arayüz geliştirilebilir ve yapının kullanılabilir olduğu sonucuna varılmıştır.

KAYNAKLAR

[1] "MYCIN | Expert System, Medical Diagnosis & Treatment", Encyclopedia Britannica. Erişim: 01 Temmuz 2024. [Çevrimiçi]. Erişim adresi: https://www.britannica.com/technology/MYCIN

[2] D. Dahiwade, G. Patle, ve E. Meshram, "Designing disease prediction model using machine learning approach", içinde *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, 2019, ss. 1211-1215. Erişim: 16 Ağustos 2024. [Çevrimiçi]. Erişim adresi: https://ieeexplore.ieee.org/document/8819782

[3] K. S. Kumar, M. S. Sathya, A. Nadeem, ve S. Rajesh, "Diseases Prediction based on Symptoms using Database and GUI", içinde *2022 6th International Conference on Computing Methodologies and Communication (ICCMC)*, IEEE, 2022, ss. 1353-1357. Erişim: 16 Ağustos 2024. [Çevrimiçi]. Erişim adresi: https://ieeexplore.ieee.org/document/9753707

[4] S. Grampurohit ve C. Sagarnal, "Disease prediction using machine learning algorithms", içinde *2020 international conference for emerging technology (INCET)*, IEEE, 2020, ss. 1-7. Erişim: 16 Ağustos 2024. [Çevrimiçi]. Erişim adresi: https://ieeexplore.ieee.org/abstract/document/9154130/

[5] "NeyimVar". Erişim: 26 Haziran 2024. [Çevrimiçi]. Erişim adresi: https://neyimvar.gov.tr/giris

[6] "Kaggle: Your Machine Learning and Data Science Community". Erişim: 24 Mayıs 2024. [Çevrimiçi]. Erişim adresi: https://www.kaggle.com/

[7] "Disease Symptom Prediction". Erişim: 24 Mayıs 2024. [Çevrimiçi]. Erişim adresi: https://www.kaggle.com/datasets/itachi9604/disease-symptom-description-dataset/data

[8] "Welcome to Flask — Flask Documentation (3.0.x)". Erişim: 24 Mayıs 2024. [Çevrimiçi]. Erişim adresi: https://flask.palletsprojects.com/en/3.0.x/

[9] "React". Erişim: 24 Mayıs 2024. [Çevrimiçi]. Erişim adresi: https://react.dev/

[10] N. Blog, "DOM (Document Object Model) Nedir, Ne İşe Yarar?", Türkiye'nin En Güncel Hosting Blogu | Natro. Erişim: 16 Ağustos 2024. [Çevrimiçi]. Erişim adresi: https://www.natro.com/blog/dom-document-object-model-nedir-ne-ise-yarar/